UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

MN

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/635,078	08/06/2003	Joseph E. Peck	5150-79600	7220

7590 03/09/2007

Jeffrey C. Hood
Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767

EXAMINER	
VU, TUAN A	
ART UNIT	PAPER NUMBER
2193	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/09/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/635,078	PECK, JOSEPH E.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 January 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-7,9-10,12-24,26-28,30-32,34-36,38 and 39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-7,9-10,12-24,26-28,30-32,34-36,38 and 39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 1/5/2007.

As indicated in Applicant's response, claims 1, 3, 9-10, 23, 26, 28, 30, 32, 34-36, 38 have been amended, and claims 2, 8, 11, 25, 29, 33, 37 canceled. Claims 1, 3-7, 9-10, 12-24, 26-28, 30-32, 34-36, 38-39 are pending in the office action.

Claim Objections

2. Claims 1, 23, 28, 32, 36 are objected to because of the following informalities: ' the first portion of the program is a successively larger portion', which amounts to a improper language usage in light of the teaching from the Specifications. A closer look at what is referred to as 'successively larger' portion reveals that it is not the case because the portion that might get larger happens to depend on the user's option, whereby occasionally one portion would happen to be increased whereas the majority of times it should get smaller; that is, according to:

Specifications, pg. 8 middle paragraph:

Thus, in one embodiment, the steps above may be repeated a plurality of times, where, in a first subset of iterations the first portion of the program is a successively larger portion of the program, and in a second subset of iterations the first portion of the program is a successively smaller portion of the program. For example, the user may move some of the (previously debugged) first portion of the program back into the remaining portion for further debugging, thereby increasing the size of the remaining portion, while decreasing the size of the first portion.

Or Specifications, pg. 11, middle para:

As indicated above, in one embodiment, user input modifying the remaining portion of the program to debug the remaining portion of the program may be received, and the method may be repeated as needed or desired. In other words, the steps described above, including those related to the test feed-through configuration, may be repeated one or more times, where generally in each iteration the first portion of the program is a successively larger portion of the program, although, as noted above, in some iterations, the first portion of the program may be (temporarily) smaller, e.g., if previously debugged parts of the program require further debugging. In other words, the debugging process described above may be performed iteratively,

where generally at each iteration the portion of the program being debugged (the remaining portion of the program) is a smaller portion of the program, but where at times, the portion of the program being debugged may increase in size, e.g., due to the user moving some of the previously debugged program portion(s) back to the remaining portion for further debugging.

Therefore, such user-driven and occasional portion would never be construed as increasing in size in a automated and programmatic fashion via successive increment (emphasis added) as claimed. There appears to be no consistent description --as to how successively becoming larger; or even smaller – that would precisely characterize or define the size behavior of this *first portion* as phrased in the above limitation. The limitation thus cannot be construed semantically as a concrete and repeatable step (i.e. whereby incremental occurs by succession). In order to provide a proper method claim (according to 35 USC § 112 first, 35 USC § 101), there should be no action step that depends on the aleatory and/or unpredictable factor played by a human intervention as disclosed (e.g. *the user may, may be temporarily, may increase in size*). This lack of a sturdy and repeatable realization of a result would characterize this recited limitation as non-concrete, rendering the use of the above language non-commensurate with the Specifications; or worse, would lead the claim to non-statutory subject matter. For lack of precise teaching and apparent overstretching language usage, the above limitation would be treated as a mere possibility to increase in size in any occasional iteration; and the Rejection would interpret the above impropriety as broad as set forth herein above.

3. The limitation recited as '...first portion of the program is a successively smaller portion' (re claims 3, 26, 30, 34, 38) also falls under a context where the 'successively smaller' phrase is not a concrete result (automated by implementation of proper method step) based on the user intervention as disclosed and identified from above. And proper reconsideration for a claim

language change is recommended. Broad interpretation will also be used to treat this uncertain character of this limitation.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1, 3-7, 9-10, 12-24, 26-28, 30-32, 34-36, 38-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Tseng et al., USPN: 6,009,256 (hereinafter Tseng).

As per claim 1, Tseng discloses a memory medium comprising program instructions for debugging a program, wherein the program is intended for deployment on a programmable hardware element (e.g. Fig. 7-8; col. 4, lines 42-50; Fig. 22-33) to perform a function, wherein the program instructions are executable to perform:

a) converting a first portion of the program into a first hardware configuration program which is deployable on the programmable hardware element to perform a corresponding first portion of the function (e.g. *hardware model 20* - Fig. 1; Fig. 6; Fig. 27-31; *maps into hardware* - col. 11, lines 47 to col. 12, line 19; step 125- Fig. 2), wherein a remaining portion of the program is to be debugged by a user (step 140 – Fig. 2);

b) configuring the programmable hardware element with the first hardware configuration program (col. 11, lines 47 to col. 12, line 19);

c) executing the program, wherein said executing comprises: the programmable hardware element executing the first portion of the program (step 125- Fig. 2); and the computer system executing the remaining portion of the program; wherein the remaining portion of the program is operable to be analyzed and debugged in response (e.g. col. 35, line 65 to col. 36, line 50) to said executing; and

d) receiving user input modifying the remaining portion of the program to debug the remaining portion of the program (e.g. col. 12, line 20-42); and

repeating a)--d) one or more times in an iterative manner, wherein for one or more iterations the first portion of the program is a successively larger portion of the program (e.g. Fig. 5; col. 15, lines 34-57).

As per claim 3, Tseng discloses: wherein for one or more iterations the first portion of the program is a successively smaller portion of the program (re claim 2 – Note: loop back into the simulation process using actual analysis results reads on successively creating larger portions of starting program; wherein each subset of actions taken per iteration reads on successively smaller portion that the whole SIM/Emualtion program – see Fig. 2, 5; col. 35, line 65 to col. 36, line 50).

As per claim 4, Tseng discloses implementing after the program has been debugged, converting the program into a second hardware configuration program which is deployable on the programmable hardware element to perform the function; and configuring the programmable hardware element with the second hardware configuration program (step 115, step 150 – Fig. 2; *step 339/YES → step 332→334, Update Registers, modeled in hardware*, step 337 – Fig. 5).

As per claim 5, Tseng discloses wherein said converting the first portion of the program into a first hardware configuration program comprises receiving user input (step 304 – Fig. 3; col. 18, line 47 to col. 19, line 13) indicating the first portion of the program.

As per claims 6-7, Tseng discloses wherein the programmable hardware element is coupled to one or more hardware resources, and wherein said executing further comprises: invoking the one or more hardware resources to perform the function (e.g. *REG1, S2, Q1* - Fig. 22-24, 26-33); wherein the program is specified to access the one or more hardware resources, and wherein the program instructions are further executable to perform:

prior to said configuring the programmable hardware element with the first hardware configuration program, analyzing the remaining portion of the program and the one or more hardware resources (Fig. 5);

determining a test feed-through configuration based on said analyzing, wherein the test feed-through configuration is deployable on the programmable hardware element (PHE) to provide for communication between the remaining portion of the program and the one or more hardware resources (Post analysis step 140 – Fig. 2 – Note: post analysis leading to step 120 reads on deployable on the PHE to iterate on more loop); and

including the test feed-through configuration in the first hardware configuration program; wherein said configuring the programmable hardware element with the first hardware configuration program further comprises configuring the programmable hardware element with the test feed-through configuration (step 332-339 – Fig. 5); and

wherein said executing the remaining portion of the program further comprises the remaining portion of the program communicating with the one or more hardware resources through the programmable hardware element (Fig. 22-33).

As per claim 9, Tseng discloses wherein said determining a test feed-through configuration and said including the test feed-through configuration in the first hardware configuration program are performed only if the remaining portion of the program is specified to access the one or more hardware resources (see Fig. 2, Fig. 5 as per rationale explained in claim 4).

As per claim 10, Tseng discloses wherein said determining the test feed-through configuration comprises modifying (step 335 – Fig 5; col. 9, lines 61-67) the test feed-through configuration based on said analyzing the remaining portion of the program.

As per claims 12-14, Tseng discloses determining the one or more hardware resources; receiving user input indicating the one or more hardware resources; querying the one or more hardware resources (e.g. Fig. 22-24, 26-33; col. 36, lines 13-40).

As per claim 15, Tseng discloses determining a plurality of pre-compiled hardware configuration program components; and assembling the plurality of pre-compiled hardware configuration program components, thereby generating the test feed-through configuration (e.g. *modeled in hardware (except memories)* -- step 332-337 – Fig. 5).

As per claim 16, Tseng discloses wherein said determining the test feed-through configuration comprises: generating a test feed-through software program based on said analyzing; and compiling the test feed-through software program, thereby generating the test feed-through configuration (e.g. col. 5, lines 45-51 – Note: providing input to a model and

simulating based on the modified model reads on compile via feed-through for generating more simulation test results – see Fig. 5).

As per claim 17, Tseng discloses storing the test feed-through configuration on the computer system, wherein the stored test feed-through configuration is retrievable for use in other reconfigurable systems (e.g. *logging selective input ... output* – col. 5, lines 40-51; col. 36, lines 20-48 – Note: recording results in log to be able to use for further testing from a point onward reads on retrievable for use by other systems – as in *bench marks* – col. 60, lines 53-62) using the one or more hardware resources.

As per claim 18, Tseng discloses determining a plurality of pre-compiled hardware configuration program components; assembling the plurality of pre-compiled hardware configuration program components, thereby generating a first portion of the test feed-through configuration; generating a test feed-through software program based on said analyzing; compiling the test feed-through software program, thereby generating a second portion of the test feed-through configuration; and combining the first portion of the test feed-through configuration and the second portion of the test feed-through configuration, thereby generating the test feed-through configuration (refer to claim 16 because of analogous sequences of steps)

As per claims 19-20, Tseng discloses a subset of the one or more hardware resources comprises one or more hardware cartridges like an I/O cartridge (Fig. 22-29 – Note: I/O data collecting entities being manipulated by user via the HDL design reads on a measurement entity -- like a gate or a register -- being not fixed and physically removed via disconnection by the modeling act of the user).

As per claims 21-22, Tseng discloses wherein the first portion of the program comprises a substantially debugged portion of the program; wherein the computer system executing the remaining portion of the program simulates execution of the remaining portion of the program on the programmable hardware element (see Fig. 2; col. 35, line 65 to col. 36, line 50 in light of corresponding mapping rationale in claim 1).

As per claim 23, Tseng discloses a memory medium comprising program instructions for debugging a program, wherein the program is usable to configure a reconfigurable system, wherein the program performs a function, wherein the reconfigurable system includes a programmable hardware element, wherein the program is intended for deployment on the programmable hardware element, wherein the program instructions are executable to perform:

a) receiving user input indicating a first portion of the program for deployment on the programmable hardware element (e.g. *hardware model 20* - Fig. 1; Fig. 6; Fig. 27-31; col. 11, lines 47 to col. 12, line 19; step 125- Fig. 2 – Note: modeling w/ HDL and RTL connectivity reads on user input), wherein a remaining portion of the program is to be debugged by a user;

b) converting the first portion of the program into a first hardware configuration program which is deployable on the programmable hardware element to perform a corresponding first portion of the function;

c) configuring the programmable hardware element with the first hardware configuration program;

d) executing the program, wherein said executing comprises: the programmable hardware element executing the first portion of the program; and the computer system executing the

remaining portion of the program; wherein the remaining portion of the program is operable to be analyzed and debugged in response to said executing; and

e) receiving user input modifying the remaining portion of the program to debug the remaining portion of the program (Note: steps b) → e) have been addressed with the corresponding mapping as set forth in claim 1);

repeating a)-- e) one or more times in an iterative manner, wherein for one or more iterations the first portion of the program is a successively larger portion of the program (e.g. Fig. 5; col. 15, lines 34-57)

As per claims 24, 26, refer to claims, 4, 3, respectively.

As per claim 27, Tseng discloses a memory medium comprising program instructions for debugging a program, wherein the program is usable to configure a reconfigurable system, wherein the program performs a function, wherein the reconfigurable system includes a programmable hardware element, wherein the program is intended for deployment on the programmable hardware element, wherein the program instructions are executable to perform:

receiving (user input indicating a first portion of the program for deployment … wherein a first remaining portion …);

converting (the first portion … a first hardware configuration program which is deployable … programmable hardware element …);

configuring (the programmable hardware element with the first …);

executing the program, (… the programmable hardware element executing the first portion of the program… computer system executing the first remaining portion of the program;

wherein the remaining portion ...receiving user input ...debug the remaining portion of the program); all of which limitations having been addressed in claim 23 above.

Tseng further discloses

receiving user input indicating a second portion of the program for deployment on the programmable hardware element, wherein the second portion of the program comprises the first portion of the program and a debugged portion of the first remaining portion of the program,

wherein a second remaining portion of the program is to be debugged by a user, wherein the second remaining portion comprises only a subset of the first remaining portion of the program;

converting the second portion of the program into a first hardware configuration program which is deployable on the programmable hardware element to perform a corresponding first portion of the function;

configuring the programmable hardware element with the first hardware configuration program; executing the program, wherein said executing comprises: the programmable hardware element executing the second portion of the program; and the computer system executing the second remaining portion of the program. (refer to claims 2-3 – Note: all of which fall under the steps repetition subject matter of claims 2-3; hence would have to be referred to claim 2 or 3).

As per claim 28, Tseng discloses a system for debugging a program, wherein the program is intended for deployment on a programmable hardware element to perform a function, the system comprising: a reconfigurable device, comprising: a programmable hardware element; and a computer system comprising a processor and a memory; wherein the computer system is

coupled to the reconfigurable device (e.g. Fig. 1; Fig. 7-8; col. 4, lines 42-50; Fig. 22-33); wherein the memory stores program instructions which are executable by the processor to :

a) convert ...; b) configure ...;

c) execute ...; and d) receive user input ...;

wherein the steps are exactly as recited in claim 1.

The steps will be rejected as set forth in claim 1.

As per claims 30-31, refer to claims 3-4, respectively.

As per claim 32, this is the system for debugging a program, wherein the program is intended for deployment on a programmable hardware element to perform a function, comprising means for performing the same steps as recited in claim 1; and these steps are rejected as set forth correspondingly in claim 1.

As per claims 34-35, these are system claims corresponding to claims 3-4, respectively; thus incorporate the corresponding rejection as set forth therein, respectively.

As per claim 36, refer to the method for debugging a program, wherein the program is intended for deployment on a programmable hardware element to perform a function, comprising:) convert ...; b) configure ...; c) execute ...; and d) receive user input ...; and repeating a)—d) as recited in claim 1 for corresponding rejection.

As per claims 38-39, these are system claims corresponding to claims 3-4, respectively; thus incorporate the corresponding rejection as set forth therein, respectively.

Response to Arguments

6. Applicant's arguments filed 1/5/07 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

Double Patenting Rejection:

(A) The amendment including the *repeating* step has been considered as to bring forth a more specific limitation; and this would presently overcome the previous Double Patenting Rejection. But this *repeating* step limitation is also subject to Objection for improper language usage; hence the withdrawal of the Double Patenting is temporary, or rather considered withdrawn pending on the correction of the above *repeating* step.

USC § 102 Rejection:

(B) Applicant has submitted that Tseng fails to teach and suggest 'converting a first program ... configuration program ... remaining portion is to be debugged ...' and '...hardware element executing the first portion ... computer ... executing ... remaining portion ... operable to ... in response to said executing' (Appl. Rmrks pg. 19, middle). The claimed steps at stakes for the discussion here amount to the following: (i) using a HW configuration program converted from a first portion program to configure a programmable HW element and have this programmable HW element run to perform a function; (ii) the remaining portion of the program (relative to said first portion) is to be debugged by an user via receiving input in order to modify said remaining portion in size in the course of repeating (i) and (ii). The cited portions of Tseng revolve about using a modeling HDL language to create RTL language constructs so to map functionality of a HW design into hardware elements, in the course of modeling a circuit design; and via mapping of RTL constructs (see Fig 6) into a Semulation tool wherein hardware acceleration engines are coupled to the software implementation which is being compiled from the RTL/HD language constructs (see col. 11, lines 47 to col. 12 line 6). Hence the combination of HDL/RTL mapping, compiling into software partition to be mapped onto accelerators (accelerators being

mapped into being equivalent to be programmable as in FPGA – see Fig. 6, 22) for a simulation in a debug context reads on what (i) is all about. The debug process can be iteratively driven until all the requirements are satisfied (see Fig 2, 5), in the context that the user can choose to go to a particular point to modify register content or change internal values (see col. 12, lines 7-20), such that this user intervention would read on (ii); that is, by exercising the user's wish to go back to one point for a rerun, the size of the software portion to be rerun can be incremented for a given stage of the debug session, wherein more iterations are usually scheduled (see Fig. 2, 5). The claim language has been interpreted as set forth above and addressed accordingly, considering the interpretation given to the impropriety of what is phrased as 'successively larger' as mentioned above in the Claims Objections. Tseng has fulfilled the claim steps; and Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(C) Applicant has submitted that nowhere is Tseng teaching about deployment on a programmable hardware element (Appl. Rmrks pg. 19, bottom; pg. 20, middle). The claim recites 'for deployment on a programmable ... element', 'which is deployable on the programmable ... element'. As recited, the act of deploying is interpreted as having code ready to be run and loaded onto the element to be executed. The fact that Tseng's FPGA are mapped into to simulate circuit components via execution and debugging does not take away the basic concept that code is loaded onto a programmable HW element, so that when running the code can help shedding insight for corrective action. The claim as a whole by virtue of (i) and (ii) in section B above amounts exactly to this paradigm: have code configured into some

programmable element, have it executed while the remaining portion is analyzed for further modification as to what can be executed or skipped; rendering the issue of comparing design endeavor and deployment endeavor moot. Every step of the claim has been interpreted and mapped with Tseng's cited portions, based on interpretation of the language as claimed. It appears as though the *deployment* concept is deemed very crucial. The claim(s) however does not contain sufficient teaching (e.g. via more concrete limitations) that would distinguish any form of claimed code execution (and debug analysis) away from the debug environment by Tseng, considering that code deployed onto a hardware element is analogous to deployment for evaluation. The argument about *deployment* is deemed non-persuasive to overcome the rejection, and the argument about deployment *being distributed between* a computer and programmable hardware element in light of what is being claimed is also non-persuasive; notably because the import of computer executing and hardware element executing is not specified in terms of novel distribution whatsoever (in the claim) to preclude the FPGA simulation and the user's approach steps in Tseng. Applicant's arguments do not comply with 37 CFR 1.111(c) because they do not clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. Further, they do not show how the amendments avoid such references or objections.

(D) Applicant has submitted that nowhere does Tseng disclose two portions on the programmable hardware element and computer systems (Appl. Rmrks pg. 21, middle; pg. 22, 2nd para). This is referred back to the analysis in section B and C. Further, it is noted that Tseng has fulfilled the first portion of the program: code being mapped into accelerators or FPGA for

execution. Tseng enable user's running of the Simulation to obtain data and re-modify that first portion via register value changes for a rerun; and this reads on remaining portion relative to the first portion; both the remaining portion and the first portion are combined to execute and/or validate the functionality of the circuit design, the main program. The claim does not make it clear how the execution of the first portion is implemented for it to be distinct in resources and hardware context/embodiment from the remaining portion computer execution: that is, interfacing and hardware communication interchange that would clearly distinguish a hardware device running separate from a computer running, so that any code moving from the remaining portion is channeled back via specific hardware interface to enable the first portion to incremented/modified, recompiled (in a separate hardware context) and rerun distinctly from the environment executing the so-called remaining portion. Lacking specifics, the first portion is interpreted as running in a FPGA while the remaining portion is residing in the user level computer executing of the S-Emulation, as set forth in the rejection.

(E) Applicant has submitted that Tseng does not teach implementation of a FPGA, but rather a model whereby modeling implementation can be intended on a FPGA as in a design endeavor not a deployment for a FPGA implementation (Appl. Rmrks pg. 23, middle). The argument revolves about imparting the weight of the term deployment versus the concept of design in Tseng. The *deployment* concept has been interpreted and analogized to just applying code into a hardware element so to have it run in order to evaluate its applicability. And the Simulation by Tseng provide the capability to attest on applicability of code being mapped and partitioned into FPGA; thus has fulfilled what entails in the concept of deploying as claimed.

(F) Applicant has submitted that the *repeating* limitation is not disclosed by Tseng (Appl. Rmrks pg. 24). It is noted that the size increase of the first portion (or moving of code into a portion to so have it successively incremented) is not given more merits than it has been indicated in the Claim Objections. And Tseng by providing alternating scenarios by which code can be rerun has fulfilled the *repeating* step; that is, user intervention providing the extra dimension or possibility to increase code size in any occasional iterative stage of the debug or SEmulation. The distinguishing of what is claimed as *remaining portion* versus *first portion of the program* has been addressed in section D above. When a debugging process repeats itself via loop back as set forth in Tseng's Fig. 5, the argument by Applicant that no *iteration* occurs would be deemed largely non-persuasive.

(G) Applicant has submitted that claim 27 has two main partitions (Appl. Rmrks pg. 25) and nowhere Tseng has disclosed this distinct partitions of steps, with emphasis on the remaining portion with respect to the first portion. Tseng by enabling the user to exercise the option to go back anywhere in the first portion and rerun a modified portion of the first has fulfilled what is construed as first portion is being executed while the computer is executing the remaining portion of said first portion. The user intervention while running the Semulation tool reads on the remaining portion while the specific and limited portion of code being mapped inside an accelerator reads on the first portion. There is nothing particularly specific about reciting remaining portion that would enforce a different scenario than that established by the rejection using the dual aspect of execution by Tseng's emulation. The argument is deemed non-persuasive in light of the broad (and particularly improper – as set forth in the Objections) language claim.

In all, the claims will stand rejected as set forth in the Office Action.

Interview Summary

7. The Applicant's representative, Jeffrey Hood, was contacted by phone in the week of February 10, 2007 in regard to reaching to a possible mutual agreement between the Examiner and the representative, so to impart some additional limitations to the paradigm of the base claims like claim 1. But no agreement has been reached; and no follow-up communication by phone has been recorded.

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tuan A Vu
Patent Examiner,
Art Unit 2193
March 8, 2007